

Preface

The promise of software reuse and the resulting rapid development of distributed business applications are becoming a practical reality through recent advances in component-based software development. The IBM SanFrancisco* product is a component-based framework for such efforts and has been supporting effective software reuse by IBM and many independent software vendors (ISVs) for over two years. During that time, the focus has shifted from the framework architecture to the practical use of this Java**-based collection of components.

This issue contains an introductory essay, four papers, and a technical note on recent progress in the SanFrancisco product and its tools, components, and applications. These contributions focus on new and recent tools, components, and applications. This issue also contains two papers and two technical notes on a variety of subjects: business function specification, an extension of the IBM Open Blueprint* supporting distributed applications, an enterprise directory service, and host emulators for use on personal computers in wireless networks. We are indebted to V. M. Johnson of the IBM Software Group in Rochester, Minnesota, for his continued planning and development of papers related to SanFrancisco and for organizing the theme on that subject in this issue.

Johnson and Austreng provide an overview of the SanFrancisco section in an essay comparing SanFrancisco today with its earlier version, described two years ago in the *IBM Systems Journal*. Not only has the product changed by becoming easier to learn and use, but the market has also changed. In addition to improvements based on early customer experience, the product now implements e-business requirements and XML (Extensible Markup Language) in support of component communication. The authors introduce the other SanFrancisco papers and technical note.

The paper by Van Emde Boas-Lubsen describes a tool for rapid development of SanFrancisco-based

applications. The tool supports both modeling and code generation and is based on an extensible architecture. The metamodel of the tool allows it to be quickly changed—in fact, the core part of the tool was developed using an earlier version of the tool itself. The version of the tool currently shipped with the SanFrancisco product is intended for marketing, evaluation, and simple prototyping use. It is not intended for developing production applications.

Bunting describes a set of tools intended for developing production applications. This tool set from METEX Systems, Inc. includes wizards for modeling and code generation. It is integrated with Rational Rose**, which provides visual modeling support, and with VisualAge* for Java and other code generation facilities. Called “Rose SF Bridge,” it allows the model and code to remain consistent as application development proceeds. The paper describes how the tool set can be used to develop, and make changes to, a SanFrancisco application.

In the paper by Jaufmann and Logan, we move from tools to a SanFrancisco-based application. The authors explain how Provider Solutions Corporation adapted the warehouse and order management components and processes of SanFrancisco for use in another domain: human resources in health care. They show how product orders relate to patient appointments and order deliveries relate to patient services, among other mappings. They also explain the development process for this application and discuss legacy code considerations.

Tost and Johnson show how JavaBeans** and Enterprise JavaBeans** components can be used to build flexible, mission-critical, SanFrancisco-based applications. A SanFrancisco-based application can be constructed as three logical tiers—the user interface (view), the application flow (controller), and the application data (model)—following a well-accepted design practice. This reduces dependencies among tiers, which may be running on different machines.

The authors show how dependencies among tiers can be further reduced with components that channel interactions between tiers.

In the technical note that closes the San Francisco section, Van der Salm reports on the progress of one company, IBS Consist B.V., in using the framework and components. The initial selection of San Francisco by the company, and its introduction into the development process, were reported in a previously published paper. The experience documented in this paper shows that the San Francisco components did indeed deliver on the promise of component-based development, while noting that this kind of development requires changes to the traditional development process.

Bevington's technical note describes a set of constructs for specifying requirements for commercial applications. Supported applications are based around one or more databases shared by many users. The constructs can be used to specify applications, databases, transactions, functions, dialogs, workflows, events, and user interfaces as well as objects and their properties, methods, and relationships. According to the author, the constructs allow a better separation of design concerns and provide more precise and complete communication between business analysts and programmers.

In the last technical note, Ricke looks at an information technology solution from the perspective of the various hardware and software products that can be used for its implementation. Building on the IBM Open Blueprint structure, he describes a three-dimensional framework, along with its architecture and a prototype implementation, that can be used to design a solution to support a business need. The framework is particularly useful for networked applications. His architecture includes the notion of a "cloud" that can be used at various levels of abstraction during the solution design process.

As special-purpose databases, directories are involved with many applications. Directory services are an important part of distributed computing, and Lightweight Directory Access Protocol (LDAP) is currently a popular protocol for these services, allowing users to easily store and retrieve data in a directory. Shi et al. describe an implementation of LDAP that uses IBM DATABASE 2* (DB2*) to implement the directory service requirements. The authors then compare the performance of their implementation to an implementation based on a file system.

Access to legacy systems will continue to be required in the near future, and host emulation is a preferred method for accessing host applications from personal computers. With mobile computing becoming more widespread, there is a greater need for emulation over wireless networks. Housel and Shields discuss Emulator Express, an IBM middleware product designed to improve the performance of Telnet 3270 and 5250 host emulation over low-bandwidth networks. They present a new caching and compression technology that greatly improves the performance of using emulation in running host applications.

The next issue of the *Journal* will be a special double issue (Volume 39, Numbers 3 and 4) on the current projects of the Massachusetts Institute of Technology (MIT) Media Lab, famous for its innovative human-computer interfaces, future-oriented advanced technologies, and other computer-related projects.

Gene F. Hoffnagle
Editor

*Trademark or registered trademark of International Business Machines Corporation.

**Trademark or registered trademark of Sun Microsystems, Inc. or Rational Software Corporation.